80/260/CDV

COMMITTEE DRAFT FOR VOTE (CDV)
PROJET DE COMITÉ POUR VOTE (CDV)

	Project number Numéro de projet	80/61162-400/Ed. 1
IEC/TC or SC: **80** CEI/CE ou SC:	Date of circulation Date de diffusion **2000-04-07**	Closing date for voting (Voting mandatory for P-members) Date de clôture du vote (Vote obligatoire pour les membres (P)) **2000-09-15**
Titre du CE/SC:	TC/SC Title: Maritime navigation and radiocommunication equipment and systems	
Secretary: M. A. RAMBAUT - United Kingdom Secrétaire:		
Also of interest to the following committees Intéresse également les comités suivants	Supersedes document Remplace le document 80/175/CD & 80/176/CD	

Horizontal functions concerned
Fonctions horizontales concernées

☐ Safety / Sécurité ☐ EMC / CEM ☐ Environment / Environnement ☐ Quality assurance / Assurance qualité

CE DOCUMENT EST TOUJOURS A L'ETUDE ET SUSCEPTIBLE DE MODIFICATION. IL NE PEUT SERVIR DE REFERENCE.

LES RECIPIENDAIRES DU PRESENT DOCUMENT SONT INVITES A PRESENTER, AVEC LEURS OBSERVATIONS, LA NOTIFICATION DES DROITS DE PROPRIETE DONT ILS AURAIENT EVENTUELLEMENT CONNAISSANCE ET A FOURNIR UNE DOCUMENTATION EXPLICATIVE.

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.

RECIPIENTS OF THIS DOCUMENT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

Titre :

Title :

Maritime navigation and radiocommunication equipment and systems - Digital interfaces - Part 400: Introduction and general principles, multiple talker and multiple listeners - Ship systems interconnection

Introductory note

IEC 61162-4 Series specifies a communication protocol for use in integrated systems. It defines a ship wide and system level integration mechanism that complements communication solutions provided by other parts of the IEC 61162 series. It is also expected that the IEC 61162-4 Series will be used for data acquisition by higher level, non real-time and non-critical administrative workstations and personal computers. IEC 61162-4 Series has been developed as a network that can support a high number of nodes (several hundred if proper segmentation is used), with response times between 0.1 second and 1 second dependent on load. Ethernet and Internet protocols are employed at the transport level.

IEC 61162-4 has been divided into four different parts numbered IEC 61162-400, 401, 410 and 420.

ATTENTION **Parallel IEC CDV/CENELEC Enquiry)**	**ATTENTION** **CDV soumis en parallèle au vote (CEI) et à l'enquête (CENELEC)**

FORM 7B (IEC)
1999-10-01

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

MARITIME NAVIGATION AND RADIOCOMMUNICATION EQUIPMENT AND SYSTEMS - DIGITAL INTERFACES-

Part 400: Introduction and General Principles Multiple Talker and Multiple Listeners – Ship Systems Interconnection.

FOREWORD

1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of the IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested National Committees.
3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical specifications, technical reports or guides and they are accepted by the National Committees in that sense.
4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
6) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

This CDV for the International Standard IEC 61162-400 has been prepared by Technical Committee 80: Maritime Navigation and Radiocommunication Equipment and Systems.

[This CDV replaces the first edition of this document circulated as a CD.]

The text of this CDV is based on the following documents:

FDIS	Report on voting
XX/XX/FDIS	XX/XX/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 3.

The committee has decided that the contents of this publication will remain unchanged until April 2003. At this date, the publication will be

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

INTRODUCTION

The International Standard IEC 61162 has been prepared by Technical Committee 80: Maritime Navigation and Radiocommunication Equipment and Systems.

IEC 61162 is a four part standard which specifies four digital interfaces for applications in marine navigation, radio-communication and system integration.

The 4 parts are :

IEC 61162-1 Single Talker and Multiple Listeners

IEC 61162-2 Single Talker and Multiple Listeners - High Speed Transmission

IEC 61162-3 Multiple Talker and Multiple Listeners - Serial Data Instrument Network

IEC 61162-4 Multiple Talker and Multiple Listeners - Ship Systems Interconnection. This part is sub-divided into a number of individual standards with part numbers in the 400 series.

This introduction is concerned with the application aspects of these four parts.

GENERAL

It is the intention of these standards to facilitate safe inter-operability and support the functionality required by modern systems and equipment, thereby satisfying the needs of ship owners, operators, manufacturers, yards and regulatory bodies. Safe interconnection may be achieved if the functionality supported herein is appropriate to the application and the specification has been properly implemented. Every effort has been made to ensure that the specifications will support both current functionality and the increasing demand for advanced functionality only realisable by integration of systems.

With regard to network communications, the specification has been separated as far as practicable into parts dependent and independent of the implementing technology, such that technological advances might be readily and safely adopted into the framework with minimal revision.

It is stressed that operational safety will ultimately depend upon the correct selection and implementation of the specifications detailed herein - safety is not (and cannot be) intrinsic to the specification. While every measure has been taken to ensure that the specifications are capable of supporting safe implementation, it remains that they will not suit every application, nor might they be considered appropriate by the regulatory body. Some guidance on applicability will be provided; however the user is cautioned to take due cognisance of any requirements imposed by the regulatory bodies.

The migration from point to point communication links (single talker / multi-listener as in parts 1 and 2 of this standard) to bus-systems (multi-talker / multi-listener as in parts 3 and 4) is to some degree driven by more extensive requirements for connectivity and easy cabling. However, this migration is also one that transforms a relatively simple system with well known failure modes to a much more complex system where failure modes are more difficult to determine and handle.

The designer needs to be aware of this problem when a new system is considered. In particular, load characteristics, response times and failure propagation will be dependent also on nodes not participating in an exchange of information. These parameters must in general be determined as a systems property.

RATIONALE FOR SPECIFIC MARINE STANDARDS

While there are a number of standardised and proprietary interconnection specifications available, there are none which provide in themselves a complete description of the services required for marine applications.

One key difference in the use of interconnection standards in the marine environment is the conceivable diversity of applications. A ship is in essence a floating community, intended to sustain both persons and cargo in often hostile conditions. General interconnection standards are not usually developed with such diversity in mind; being fundamentally limited in their scope, and hence their application. This IEC Standard 61162 provides four specifications to support the services required for marine applications: decision support, data acquisition, shipboard and safety management, etc., within the framework and constraints imposed by the various regulatory bodies.

The adoption of proprietary or generalised industrial specifications in the marine environment can in itself pose risks in implementation. Deviation from a generalised specification to support marine specific requirements could potentially lead to the introduction of systematic faults. Furthermore, such specifications may be intended to support functionality not relevant to marine applications, leading to the inefficient use of often limited resources. It should also be considered that dependability issues such as availability, reliability and maintainability often have safety as well as commercial implications in the marine environment, e.g. the loss of steering or propulsion.

With regard to hardware, components manufactured for more benign or general industrial applications may not be suitable for the often hostile marine environment, or be accepted as such by regulatory bodies. It is therefore intended that implementation of this standard need not depend on “commercial off the shelf” (COTS) technologies. This rationale is equally applicable to software development, since the acceptability of such may also depend on the application and imposed requirements. A further advantage of this rationale is that the standard will continue to support performance standards adopted by IMO and other marine regulatory bodies, rather than be compromised by the demands of other potential users.

The standard will offer benefits to system developers in that economies of scale might be made by reducing application specific development, facilitating common hardware and software platforms and eliminating the need to individually specify or adapt other specifications. Yards will benefit in that the installation and testing of disparate systems will be simplified considerably. Owners and operators will benefit from better integrated systems capable of implementing advanced functionality, rather than ad hoc solutions. Regulators will benefit from the application of a cohesive and systematic standard, which readily supports verification and validation.

The following sections will provide an overview of the characteristics of each of the four specifications, with a view towards clarifying the applicability of each. The user of this standard may therefore be guided towards an appropriate selection for the given implementation.

IEC 61162-1 Summary

This standard is intended to support one-way serial data transmission at 4800 bits/s from a single Talker to one or more Listeners. This data is in printable ASCII form and may include information such as position, speed, depth, frequency allocation, etc. The number of characters in a message may be from 11 to a maximum of 71 and generally the message is not required to be transmitted more often than once per second. The sentence length is 82 characters taking account of the overhead of 11 characters.

Detailed definition of all transmitted data is included and a means of accommodating proprietary data is provided.

Interconnection between devices may be by means of a two-conductor, shielded, twisted pair cable. No provision is made for more than a single Talker to be connected to the line.

Multiple Listeners may be connected to a single Talker. The Listener’s receive circuit shall consist of an opto-isolator to provide ground isolation.

The electrical definitions in the standard are not intended to accommodate high-bandwidth applications such as radar or video imagery, or intensive database or file transfer applications.

There is no provision for guaranteed delivery of messages and only limited error-checking capability.

IEC 61162-2 Summary

The electrical definitions in this standard are intended to accommodate higher data rates than are specified in IEC 61162-1. This standard IEC 61162-2 is intended to support one-way serial data transmission at 38400 bits/s from a single Talker to one or more Listeners. This data is in printable ASCII form and may include information as specified by approved sentences or information coded according to the rules for proprietary sentences. The number of characters in a message may be from 11 to a maximum of 71 and generally the message is not required to be transmitted more often than once per second. The sentence length is 82 characters taking account of the overhead of 11 characters.

Detailed definition of all transmitted data is provided by the IEC 61162-1 standard.

Interconnection between devices may be by means of a shielded two-conductor twisted-pair wire (A,B) plus any means to secure common signal ground potential (C) for transmitting and receiving devices. For this purpose a third wire, additional to the twisted pair or inner shield of a double shielded cable with insulated shields, may be used.

No provision is made for more than one Talker to be connected to the line. The Listener receive circuit shall provide galvanic isolation. Multiple listeners may be connected to a single Talker. The improved electrical interface specification permits the connection of 10 or more Listeners.

There is no provision for guaranteed delivery of messages and only limited error-checking capability.

IEC 61162-3 Summary

This standard is designed to support bi-directional data communication between multi-talker and / or multi-listeners at a speed of 250 kbit/s.

IEC 61162-3 contains the requirements for the minimum implementation of a serial-data communications network to interconnect marine electronic equipment onboard vessels. Equipment designed to this standard will have the ability to share data, including commands and status, with other compatible equipment over a single signalling channel.

This standard is based on the CAN specification (ISO 11898) which limits the non-fragmented message size to 8 bytes. This also provides fragmented “fast packet” messages of up to 223 bytes and a “multi packet” messages of up to 1785 bytes. Data messages are assigned unique priorities, each are allocated priority and transmitted as a series of data frames, each with robust error checking and confirmed frame delivery. The standard does not, in itself, support physical redundancy

The communication speed is 250 kbits/s, corresponding to a cable length of 200 m.

Provision is made for the interface circuits to be powered from the bus and galvanic isolation is necessary between the bus and devices in the system.

This standard is not intended to support high-bandwidth applications such as radar, electronic chart or other video data, or other intensive database or file transfer applications.

IEC 61162-4 Series Summary

This standard is intended to support bi-directional data communication between multiple talkers and/or multiple listeners at a speed greater than 10Mbits/s, in order to facilitate interconnection of ship-borne systems.

IEC 61162-4 Series specifies a communication protocol for use in integrated systems. It defines a ship wide and system level integration mechanism that complements communication solutions provided by other parts of the IEC 61162 series. It is also expected that the IEC 61162-4 Series will be used for data acquisition by higher level, non real-time and non-critical administrative workstations and personal computers. IEC 61162-4 Series has been developed as a network that can support a high number of nodes (several hundred if proper segmentation is used), with response times between 0.1 second and 1 second dependent on load. Ethernet and Internet protocols are employed at the transport level.

IEC 61162-4 specifies a system interconnection protocol on a higher architectural level than the other parts of this standard. Lower level protocols (-1 to -3) should be used for direct data acquisition and control within time critical loops.

The -4 standard supports the transfer of messages and streams. Messages can be delivered through broadcasts or as point to point .Streams are supported to facilitate transfer of larger amounts of data, e.g., ECDIS charts, RADAR images or other bulk data. Streams are always point to point. Messages up to 1400 bytes can be transmitted without fragmentation, longer messages will be supported as multi-fragmented transmissions.

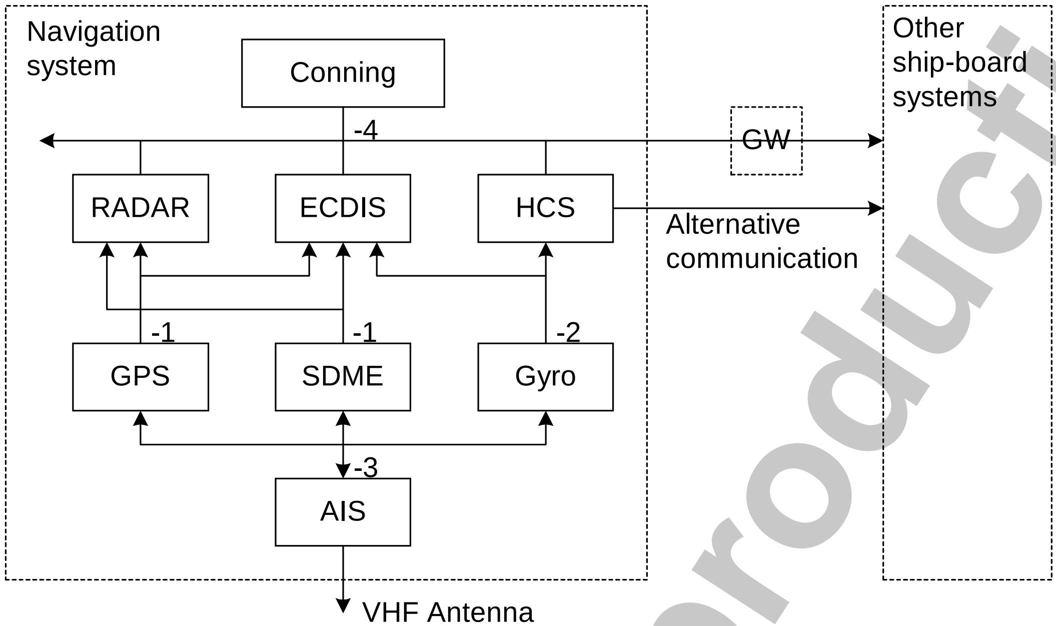
The capacity of the IEC 61162-4 network is dependent on the number of nodes in the network, the hardware and software in use and the actual physical network layer. As a minimum the standard will provide 10 Mbit/s shared by all nodes on the network, significantly higher capacity is possible with appropriate network technology. Actual performance must be determined by system analysis or measurement.

APPLICABILITY OF THE DIFFERENT STANDARDS

The different standards in this series have different uses. The following table illustrates important differences between the standards, from which the applicability can be evaluated.

#	Application	61162-1	61162-2	61162-3	61162-4
1	Data Repetition Rate	7 Hz	50 Hz	7k Hz	10 Hz
2	System bandwidth	4.8 Kbits/s	38.4 Kbits/s	250 Kbits/s	10 Mbits/s
3	Number of Listeners	1	10	50	100
4	Number of Talkers	1	1	50	100
5	Message fragment length	80 byte	80 byte	8 byte	1400 byte
6	Cable length	500m	500m	200m	500m

A typical ship control system may make use of all these standards as the following figure shows. The series 4 standards are typically used for high level integration while other parts of the standards are used in lower level and more time critical components. The figure shows an example system implementation and is in no way normative.



The various communication protocol standards are shown as solid lines with a label indicating the type. The dashed lines show the possible segmentation of the complete integrated ship system into several components, here a navigation system and other systems, containing, e.g., the rudder control system.

The use of the different standards illustrated here, is based on the following rationale:

High level integration is done with the part 4 standards. This standard may also be used in other systems, possibly via some form of gateway (GW).

Part 1 and 2 standards are used in point to point links between navigational equipment mainly. The part 2 standard may be used in higher speed applications (e.g., from the Gyro) than what is possible with the part 1 standard.

The part 3 standard may be useful to collect large amounts of data from many sensors in a very time critical or cost sensitive environment. Here it is illustrated as used to collect data for the Automatic Identification System (AIS).

Additional (proprietary or other open) communication protocols may still be used in special cases. here it is illustrated with a direct link (analog or serial) from heading control device (HCS) to, e.g., rudder machinery.

CONTENTS

Page

1 Scope

1.1 General

This standard series, IEC 61162-400 and upwards, specifies a communication protocol for use in integrated ship control (ISC) systems. It also specifies an interface description language for use together with the protocol, a set of rules for the use of this language and a set of standard interfaces described in the language. Finally, it also provides a test plan and list of required documents for equipment using this standard.

This part of the standard gives a general overview of the functionality of the protocol and provides definitions common to the other fragments of the standard.

1.2 Application area

This protocol is intended for use on the system level of an integrated ship control and monitoring system. It is designed to integrate various relatively large functional components, e.g., RADAR, ECDIS or conning display. As such, it complements other protocols on the instrument level (IEC 61162-1/2/3 as referred to in the preface) and on the administrative level (mainly proprietary or de facto standard protocols).

Although this standard covers navigation and radiocommunication equipment on the system level, it is not limited to that. It could also find application on lower levels (process level) and in other application areas (general automation).

1.3 Safety implications of using this protocol

This standard does not define any safety related attributes that can be applied in the verification of the safety properties of a system using this protocol. The system safety properties will dependent on many factors, such as.:

a) The protocol specification (this standard).
b) The T-profile in use (may be specified by this standard).
c) The protocol implementation (dependent on manufacturer).
d) How the protocol is used by individual components (dependent on manufacturer).
e) How the system use the protocol (dependent on manufacturers and system integrators).
f) Maintenance and supervision of the system.

These items are only examples and do not constitute a complete list. The relevant authorities and the class societies will prescribe more detailed rules for the use of this protocol in integrated control systems.

1.4 Components of this standard

This standard consists of a number of documents (parts). This introduction contains a general description of the functionality of the standard and guidelines for the use of the other documents. The relationship between documents are indicated in the below figure.

Although this set of standard documents is collectively referred to as IEC 61162-4, the actual part numbers are in the 400-series. The part numbers are shown in the figure.

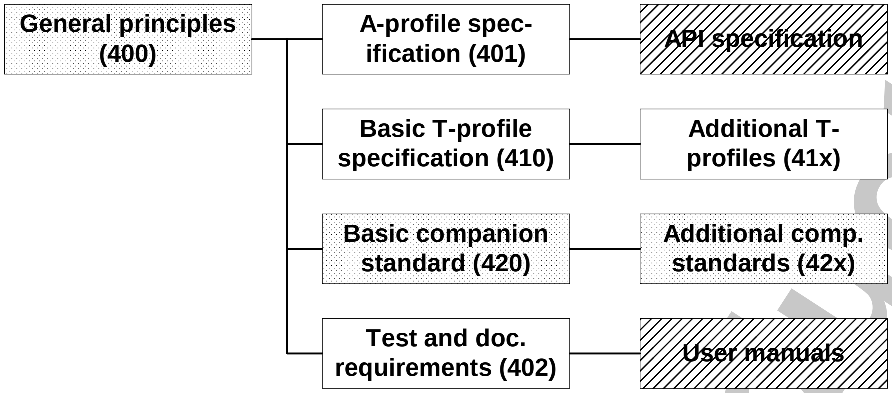


Figure 1 - Relationship between standard documents

The documents marked with a diagonal line pattern are not part of the standard. They are required programmer or operators manuals provided by manufacturers of equipment or components using this standard.

The non-shaded documents are documentation required for designers of communication libraries implementing this standard. They are not required for manufacturers of equipment using existing communication libraries.

The companion standards documents (shaded) are required reading for designers and integrators of equipment using this standard. They are also of interest to those who specify equipment for ships.

The general principles are required reading for all users of the standard. The general principles give a high level explanations to the various parts as shown in the below table.

Table 1 - Parts of general principles document

Clause	Contents	Required for part
Scope	Purpose and overview	All
Overview and general principles	General description of application area and usage	All
T-profile functionality	General description of requirements for implementation of this protocol on top of specific transport service	IEC 61162-401 A-profile IEC 61162-410 T-profile
A-profile functionality	General description of functionality of application level protocol	IEC 61162-420 Companion standard general principles, IEC 61162-401 A-profile
Companion standard functionality	General description of Purpose and functionality of companion standards	Companion standards, Application descriptions.
System configuration services	Requirements for integrating systems using this protocol	IEC 61162-401 A-profile IEC 61162-410 T-profile

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All normative documents are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 7498: 1984, Information processing systems – Open Systems interconnection – Basic Reference Model.

ISO 8859: 1987, Information processing – 8-bit Single Byte Coded Graphic Character Sets.

ISO 8859-1: 1987, Part 1: Latin Alphabet No 1.

IEC 61162-1: 1995, Maritime navigation and radio communication equipment and systems - Digital interfaces - Part 1: Single talker and multiple listeners

IEC 61162-2: 1998, Maritime navigation and radio communication equipment and systems - Digital interfaces - Part 2: High speed single talker and multiple listeners

IEC 61162-3: (To be published), Maritime navigation and radio communication equipment and systems - Digital interfaces - Part 3: Multiple talker and multiple listeners – Serial Data Instrument Network

IEC 61162-4: (short-hand for all parts in the 400 series – to be published), Maritime navigation and radio communication equipment and systems - Digital interfaces - Part 4xx: Multiple talker and multiple listeners - Ship Systems Interconnection.

IEC 61162-401: (to be published), Maritime navigation and radio communication equipment and systems - Digital interfaces - Part 401: Multiple talker and multiple listeners - Ship Systems Interconnection – Application Profile.

IEC 61162-402: (to be published), Maritime navigation and radio communication equipment and systems - Digital interfaces - Part 402: Multiple talker and multiple listeners - Ship Systems Interconnection – Documentation and Test Requirements.

IEC 61162-410: (to be published), Maritime navigation and radio communication equipment and systems - Digital interfaces - Part 410: Multiple talker and multiple listeners - Ship Systems Interconnection – Transport Profile Requirements and Basic Transport Profile.

IEC 61162-420: (to be published), Maritime navigation and radio communication equipment and systems - Digital interfaces - Part 420: Multiple talker and multiple listeners - Ship Systems Interconnection – Companion Standard Requirements and Basic Companion Standards.

3 Definitions

For the purposes of this International Standard the following definitions apply:

A-profile
Communication protocol supplying application services (see *OSI* 5 to 7).

ABC - Anonymous Broadcast (MAU)
A mechanism by which a MAU can send or receive data with no defined peer or group of peers.

Accept
The term accept (or server) is used to define the *MAU* (or other entity associated with it) that has exported a *data object*. The term client (or connect) is used about the *MAU* (or other entity associated with it) that use the *data object*.

API - Application Programmer's Interface
One implementation of the required application services as defined in IEC 61162-401. One API from one manufacturer may be different from another API, although the basic functionality is the same.

Bridge (in the context of a data network)
A network bridge is used to connect two or more network segment together. It will normally do this on the data-link level, i.e., it will be able to isolate traffic internal to one segment from other segments, but it will not be able to perform more advanced filtering required for, e.g., fire-walls.

Callback
A subroutine in the application program called from a service provider library as a result of a previous service request.

Character
A character is an octet containing a code from the set defined in ISO 8859-1. The null character (octet containing all zero bits) may have special meaning.

Client
A client (*connect* type entity) uses the services of an *accept* type entity.

Companion standard
The A-profile part of this standard defines a protocol for transport of data structures between nodes in an integrated ship control system. It does not in itself specify how to interpret these data structures, i.e., if it is a temperature measurement or a rudder angle. The interpretation of the data objects are defined by additional documents called "companion standards" or user layer specifications.

The Companion Standards Requirements part of this standard defines rules for the creation of companion standards and how to implement them. This part also defines some general companion standards, e.g., a mapping of IEC 61162-1 telegrams.

Connect
A connect type entity (*client*) uses the services of an *accept* type entity.

Data marshalling
This standard defines a transmission format for data records that is independent of computer architecture, network particulars, compilers and programming languages.

Data marshalling routines convert between this transport format and internal data representations used in different modules.

Data object
This standard is based on a simplified object oriented client-server model. The term data object will be used in this standard to denote a logical entity that are characterised by the following:

- A data object has exactly one server (one logical network node - MAU). The object comes into existence when the information about the object is exported to the network by the server.
- A data object has zero or several clients (MAUs). Clients can connect to the object when the object comes into existence.
- Exactly one client-available operation is defined for the object.
- The defined operation can be used by the client to be informed about state changes in the object and/or inflict state changes on the object.

The data object is a “virtual“ object. The server does not export the data object to the network, it exports the identity of the object which points back to a physical data structure in the server. All operations on the object are performed locally by the server, the network will transfer information about these operations.

Fire-wall (in the context of a data network)
A fire-wall is a device connecting two or more network segments together while performing certain safety related functions. These functions are, as a minimum, to limit the load from the fire-wall onto certain of the segments and do message filtering to ensure that only a specified sub-set of functions are made available from certain of the segments.

LNA - Local Network Administrator
The LNA is the protocol processing module that interfaces the application unit (*MAU*) to the network. Each MAU has one LNA.

MAPI - MAU API
The MAPI is a generic term for an *API* that allows an application program (MAU) to interface to its LNA.

Message
A message is a fixed format sequence of octets that are exchanged between modules in an IEC 61162-4 system. All messages will be identified with a message code. All messages, their codes and formats are described in the A-profile part of this standard.

MAU - MiTS Application Unit
Historical name (see *MiTS*).

MCP

MiTS - Maritime Information Technology Standard
Predecessor to this standard. Now superceeded as a specification by the IEC 61162-4 series of standards.

MTU - Maximum Transmission Unit
Longest message length over a given T-profile.

Octet
An octet is an eight bit data entity (or termed a byte). An octet is the smallest transmission unit that is discussed in this standard. Any data entity transported over the network will consist of an integral number of octets.

Open System Interconnect (OSI)
This standard makes references to the ISO/OSI standard reference model for open systems interconnection [ISO 7498], but it does not adhere to that standard with regard to the exact services provided. The ISO/OSI standard is sometimes used as reference for the naming of the individual layers in the protocol stack (see Figure 2).

The following conventions apply:

- With respect to functionality, the protocol definitions cover the session, the presentation and the application layers of the OSI model (the *A-profile*).
- The protocol require a set of transport services. The services can possibly be supplied by any number of different transport protocols stacks (*T-profiles*). The standard transport protocols will be defined in Part 410 of this standard.
- This standard does not describe the A-profile as layered. This standard merges all the upper three layers of the ISO/OSI model into one protocol.
- This standard refers to the companion standards or user layer as a distinct protocol layer on top of the application layer.

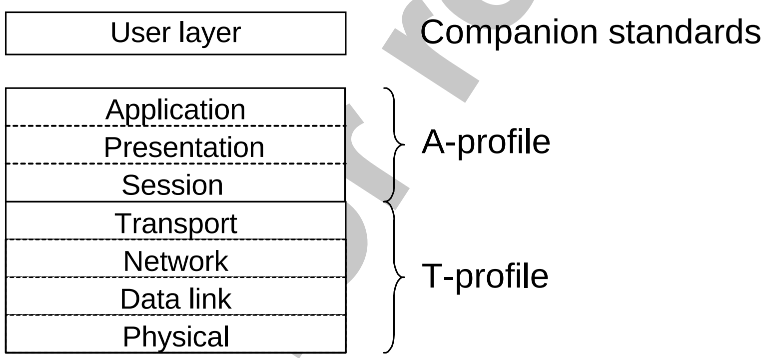


Figure 2 - Protocol layering

PFS - PISCES Foundation Specifications
Standard set of companion standards, used to ensure a certain degree of vendor independent interoperability.

PISCES
Another historical name for this standard (see *MiTS*).

Server
A server is an accept type entity (see *accept*).

T-profile
Protocol supplying transport services to the *A-profile* (see *OSI*, layers 1 to 4).

Transaction
The exchange of information between client and server in conjunction with an operation on a *data object* is called a transaction. Transactions follow a general pattern where the client

requests an operation on the object, the server performs it and the client is notified of the result. Transaction types are defined where the first or the last part is missing, i.e., where the client does not initiate the transaction or where the client does not get information on the final result.

User layer
The user layer in the OSI model represents what is called the *companion standard* in this standard.

QoS - Quality of service
To what degree a certain protocol (typically T-profile) supplies certain services to the higher levels. In the context of this standard, the QoS attributes of most interest are redundancy in transport paths, message priority and the probability that messages are delivered as specified (e.g., without errors and loss).

4 Overview and general principles

4.1 Introduction

This section gives an overview of the concepts and the general principles used in the IEC 61162-4 standard. It is an introduction to the more technical components of the standard and gives an overview of where they are defined.

4.2 Basic protocol functionality

The protocol functionality is adapted to the requirements for high level integration of ship control equipment. Clause 5 defines the application level functionality provided by this protocol.

a) Support for soft to firm real-time applications. The protocol is not appropriate for hard real-time data transport. The protocol provides support for different priority assignments to different data channels.

b) High inert safety. Error states in the network or in other nodes shall not propagate to the network or other nodes.

c) Medium active safety. The degree to which one can rely on the protocol's QoS will be dependent on the T-profile in use.

d) Support for message based traffic (e.g., control, supervision, data acquisition) as well as bulk transfer (e.g., RADAR or ECDIS images). Differentiated quality of service provides some independence between the two transport modes.

e) Support for commonly used transport primitives (remote read, write, function calls, subscriptions and broadcasts).

f) Definition of standardised data marshalling for unambiguous transfer of information between application written in different languages, on different hardware platforms or under different operating systems.

g) Client-server architecture where systems can be extended incrementally by adding new clients on top of existing servers.

h) Mechanisms for definition of unambiguous description of equipment interfaces which support automatic code generation.

i) Support for automatic and distributed configuration of network nodes. This can be used to provide plug and play service (incremental construction of control networks) or higher robustness network through automatic configuration or reconfiguration.

j) Support for network management and supervision (part of T-profile).

k) Support for application management and supervision (through A-profile).

l) Support for maintenance of a system-wide time base (part of T-profile).

4.3 Program modules

4.3.1 Physical modules

This standard specifies a system where most of the communication facilities are implemented in an application independent module called an LNA. The application module (MAU) communicates with the LNA through a simple point to point link. The different entities are shown in the Entity Relationship (ER) -diagram below.

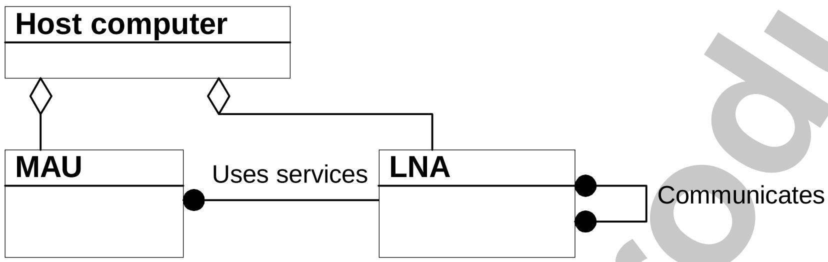


Figure 3 - Physical components

The most characteristic aspect of the modularization is that the application unit (MAU) is separated from the network interface module (LNA). This separation means that the application independent part of the system (LNA) has an opportunity to supervise and control the application module with reduced danger of error propagation from network to application and vice versa.

The separation can be through a communication link (e.g., TCP/IP), through shared memory or through a software library interface. The quality of the separation will determine the quality of the inert safety properties.

Each host computer (physical node on the network) can have zero or more MAUs and zero or one LNA. The MAU uses the services of exactly one LNA to communicate with other MAUs (through their LNAs). Each LNA can serve any number of MAUs on or off the LNA's host computer.

The MAU is the container for application programs. It uses the communication services provided by the LNA to set up communication links with other MAUs. The LNA is the application independent communication manager.

Normally the MAUs and the LNA are separate processes with separate contexts, even when running on one host computer. However, other configurations are possible:

- The LNA runs on another computer than the MAU. This is supported by this standard by using a dedicated point to point communication link between MAU and LNA. This protocol is described in the A-profile specification.
- The MAU and LNA may be real-time tasks on an embedded processor, sharing the same memory and resource context (normal for many real-time operating executives).
- The MAU and LNA may be merged into one process or task sharing both memory and general execution context. This can be a useful configuration for smaller systems where efficiency and low overhead is of importance.

The different possible configurations are organised in conformance classes as defined in 4.3.3.

4.3.2 Protocol types

The division of the module into an application part (MAU) and a network interface part (LNA) requires the use of two communication protocols:

- The MAU-LNA protocol that is mainly used within a host computer and are normally implemented as a inter-process communication link, e.g., a Windows DLL or UNIX shared memory. It can also be implemented as procedure calls in a monolithic LNA-MAU. This protocol is documented in the A-profile specification and in manufacturers' documentation. It will not be further discussed in this Part 400 of the standard.

- The LNA-LNA protocol that is used on the physical network, connecting the host computers and their LNAs together. This protocol is defined in the A-profile and the T-profile specifications. The T-profile provides low level transport services while the A-profile provides the application related services, based on the T-profile services.

The A-profile services are detailed in clause 5.The T-profile services are detailed in clause 6.

4.3.3 Protocol conformance classes

The communication link between MAU and LNA and the different possibilities in how to integrate MAU and LNA gives rise to four conformance classes. Based on how the modules are placed on different host computers, the conformance classes can be illustrated as in Figure 4

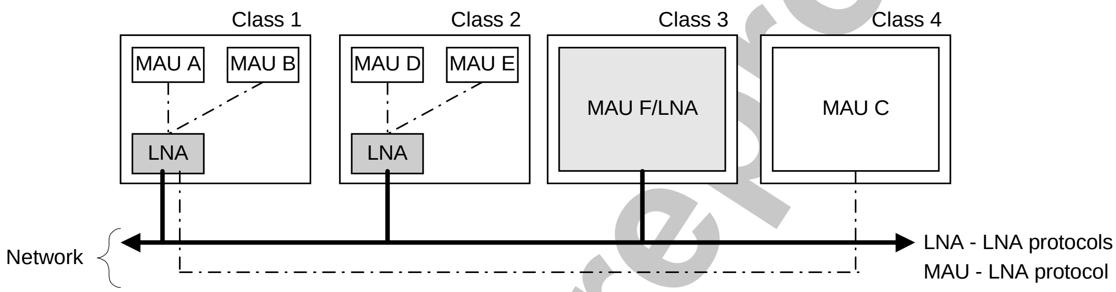


Figure 4 - Conformance classes

Four class levels of conformance for an application on a host computer are defined:

Class 1.The application contains an LNA that can accept connections from other MAUs on the same host computer (potentially delivered as a software module) and also MAUs on other host computers (conformance class 4) via the A-profile TCP/IP MAU-LNA protocol. This conformance class shall also be compatible with older (V3.1) versions of MiTs application units on the TCP/IP MAU-LNA link (see next clause).

Class 2.The application contains an LNA that can accept connections from other MAUs on the same host computer (potentially delivered as a software module), but not from MAUs on other host computers (see conformance class).

Class 3.The application is an integrated MAU/LNA that cannot accept connections from other MAUs on same or other host computers.

Class 4.The application has no LNA and is dependent on connection to an application of conformance class 1 (via TCP/IP) or 2 (via another defined Inter Process protocol IPC).

4.4 Compatibility with MiTs versions of the protocol

This version of the protocol is not compatible with older versions. The Internet T-profile defined in IEC 61162-410 is, however, specified so that applications and host computers using the previous versions of the protocol can coexist with the new version. By integrating one old and one new application on the same host computer, a gateway between the two protocols can be devised.

The application programmers interface is similar between the previous and the current version of the A-profile. IEC 61162-401 describes the differences.

4.5 API versus protocol

Part 401of this standard defines a set of services that shall be provided by the API. Different APIs will provide these services in different ways, depending on, e.g., programming language, operating system and programming paradigm. The actual definition of how to use these API services shall be documented by the designer of the API library.

4.6 Protocol level entities

Part 401 of this standard specifies several entities that are used to implement the communication services. These are identified in the diagram below.

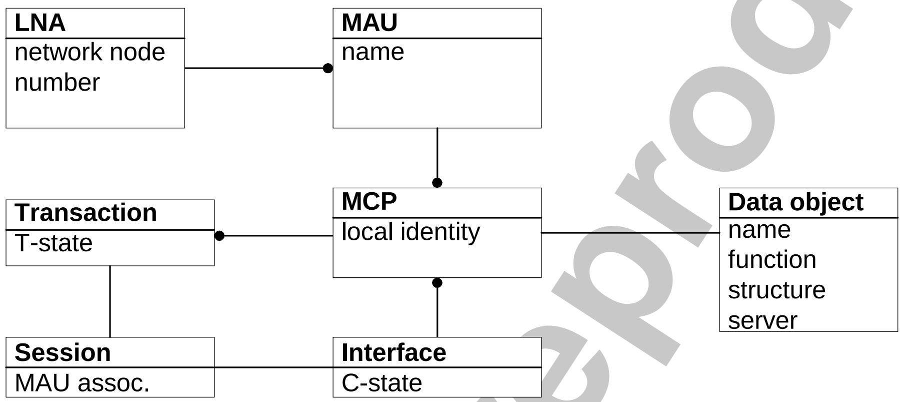


Figure 5 - Protocol entities

The **LNA**s are the physical communication nodes. They act as switches for data traffic between the MAUs they serve and remote LNAs with their MAUs. The network level address will be assigned to an LNA.

The **MAU** is served by exactly one LNA and no MAU can exist without a corresponding LNA. Several MAUs can be served by one LNA. Each MAU has a unique name. The name is as a minimum unique for the LNA, but will normally also be unique in the network. One application (a MAU) that is moved from one LNA to another will for this protocol be treated as the same MAU. This also means that two MAUs with the same name cannot be distinguished by the protocol.

Each MAU normally has a number of **MCP**s (MAU Connection Points). An MCP is a local reference to a global **data object**. Each data object is defined and served by exactly one server MAU. Any number of client MAU (also including the server MAU itself) can make use of the data object. Thus, each data object represents one component of the server MAU functionality, much as a named function in a program library does. A data object has a name and it can support exactly one operation (read, write, subscribe etc.) with one strictly defined input and/or output data structure.

The server MAU has one MCP for each data object it has defined. This MCP is shared between all clients that use this data object. Each client needs one MCP for each data object it wants to use. The server MAU accepts connection and exports the data objects to the network. The data object is an abstract entity that represents the functions that can be invoked in the network. Conceptually, this can be looked at as if the server defines the data objects and the clients connects to them. Once the connection is established, the server and client can exchange messages (perform a **transaction**). The connection is established based on identical values for the following attributes: Server MAU name, MCP name and format string (i.e., function and input/output structure).

Data objects are collected in named interfaces by the server MAU. The clients can connect to them as a complete block or as a sub-set of the block. The connection state (C-state) for the individual MCPs is associated with the interface. The interface will also be linked with a session which is a MAU-MAU association code that can be used for transaction and connection authentication. All incoming messages (except those of the broadcast type) can be traced back to their originator MAU by examining the session the transaction is associated with.

NOTE – The terms data object and MCP may appear intermixed in the text. However, in these cases it should normally be obvious from the context which meaning the term actually has.

4.7 Dependencies of actual API implementations

The services described in the following clauses are a minimum set of services based on the defined MAU-LNA messages. Most API implementation will extend the service set for, e.g.:

a) More structured handling of object call-backs, e.g., mapping the remote client MAU session closed event to a set of interface client down events.

b) A more object oriented approach, e.g., by having MCP attributes implemented as actual attributes of programming language level objects.

c) Higher level services, e.g., complete definition and establishment of MCPs or interfaces in one call.

This means that the services must be considered as minimum services and more an explanation of the protocol description than an actual API specification.

4.8 Companion standard entities

The companion standard documents allows a designer to specify attribute values for various protocol level entities (see previous clause). In addition, the companion standard will structure the use of the entities to develop a general system in the construction of applications and in how applications communicate.

There are four different types of companion standard document entities as shown in Figure 6.

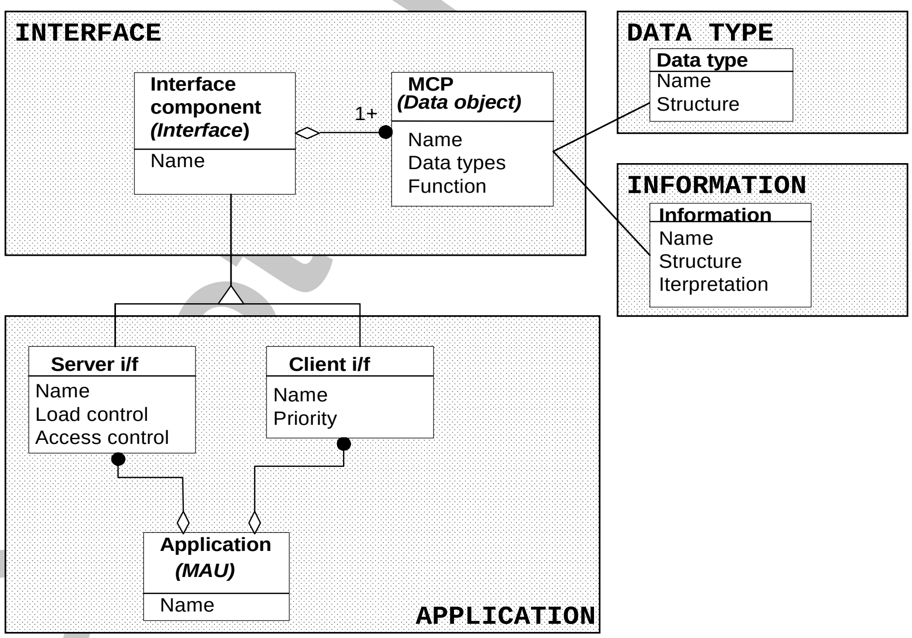


Figure 6 - Companion standard entities

The MAUs are described in the Companion Standard Application Specifications, 61162-420.

The Interfaces are described in their Companion Standard Interface Specifications, 61162-401.

The MCPs or data objects are described as a connection points also in the Companion Standard Interface specifications. Data types and information containing entities are defined in Data Type specifications, 61162-nnn.

This structuring is done by the following mechanisms:

a) Data objects and interfaces will be used as defined for the protocol level: Interfaces will consist of a group of data objects. Data objects and interfaces are defined in (component) `INTERFACE` documents.

b) Interfaces will be built from component interfaces (smaller collections of data objects) so that the client side of an interface may consist of fewer data objects than the server side and different servers may implement different sets of interface components.

c) The data structure of data objects will be defined by separate entities called `DATA TYPE` definition or by entities called `INFORMATION` definitions.

d) The companion standard may allow several identical interfaces to be implemented on one MAU. To enable the protocol level to distinguish between the interfaces, each interface can be assigned a new name by the companion standard. This is done in the `APPLICATION` definition document.

e) The companion standard will allow the definition of load and access control attributes during the aggregation of interfaces in a MAU. It will also define the name of the MAU. This is done in the `APPLICATION` definition document.

f) The companion standard will also define a hierarchy of application specifications together with a similar hierarchy of interface specifications, data structure and information entities, called the PISCES Foundation Specifications (PFS). This system will ensure a classification of applications according to functional capabilities. As part of this, there will be a set of interfaces supporting independent operations (e.g., retrieve version numbers, manufacturer’s code etc.).

4.9 Relationship between specification components and products

The various parts of the standard specification and the relationship between them are illustrated in Figure 7.

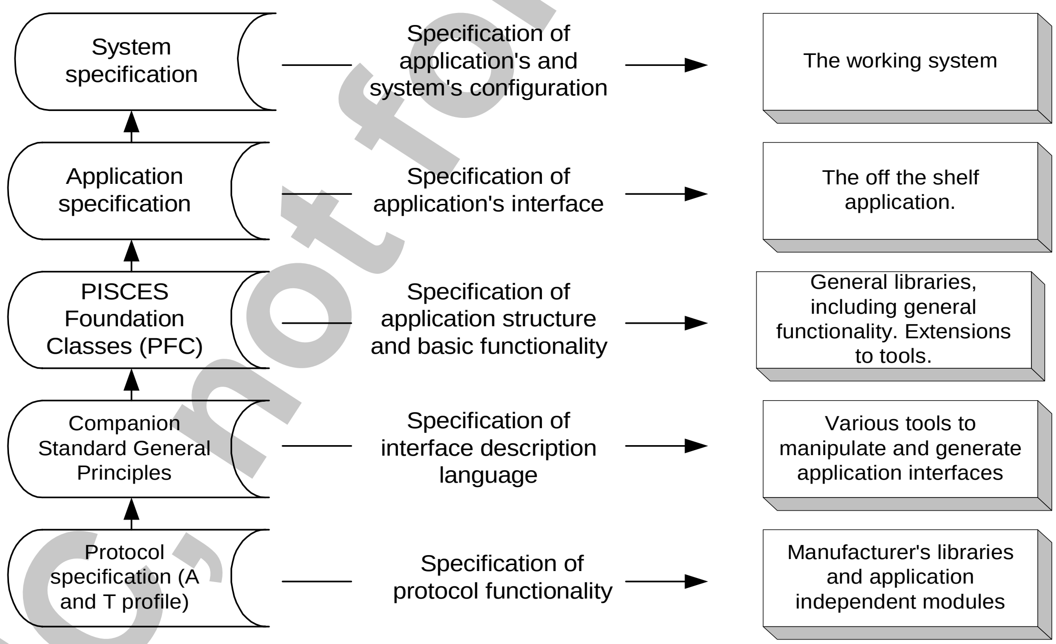


Figure 7 - Relationship between specifications and products

5 A-profile functionality

5.1 Introduction

This clause gives an overview of the contents of the A-profile specification. The A-profile is the definition of the services provided to the applications (via the MAU-LNA communication link) and a specification of how these services shall be implemented with a set of LNA-LNA messages.

The services and the protocol can be divided into the following groups:

a) Application management. Connection of application to network and monitoring and control of application's connection state.
b) Data object connection management. Exporting server data objects and connecting to them with client objects. Monitoring and control of connection state.
c) Data transfer between MAUs. Message and stream based data transfer between MAUs.

5.2 General principles

5.2.1 Separation between applications

The MAUs are the application dependent parts of the system. The LNAs are completely application independent. By providing various interface checks between MAU and LNA, it is possible to give a certain guarantee that application errors do not propagate to the network or vice versa.

The LNA will also provide certain application independent watchdog services to the MAU.

5.2.2 Automatic configuration

A MAU shall be able to connect to or disconnect from the network at any time. A new MAU shall be able to take over the functionality from an old MAU without having to reconfigure other MAUs using the services of the exchanged units.

This principle is based on having all local configuration information in each MAU. The MAU name is defined to be unique in the system. By shutting down one MAU with name "X" and then starting a new (and identical with respect to the interface it has to the network) MAU with the same name "X", all clients of the old MAU will change their connections to the new one.

5.2.3 Client-server architecture by the use of data objects

This standard prescribes a client-server architecture based on data objects. A data object is an abstract entity existing in the network. The data object is defined once a server MAU exports its definition to the network. One or more client MAUs can then connect to the object and exchange messages with the server MAU. A MAU can be both client and server to the same or different data objects.

NOTE - There is a definition of a number of "pseudo-objects" for broadcast destinations. These destinations do not have one specific server MAUs, but can be listened or transmitted to by any MAU.

5.2.4 Connection oriented

The system is connection oriented. This means that exceptional situations relating to the exchange of information between two MAUs (e.g., communication errors, death of one MAU or the destruction of one data object) will be reliably detected and can, if necessary, be handled separately from the normal transactions. This applies also to broadcast type data objects where a connection between client and server is maintained although the message transport is connection less (based on a broadcast service in the T-profile).

NOTE - Anonymous broadcast subscribe type objects are not connection oriented.

5.2.5 Transaction oriented

Most communication between MAUs is done as client initiated transactions. The client establishes a connection to an object (previously defined by the server MAU) and can, once the connection is established, initiate transactions on the connected object. For each transaction request the client will get from zero to an "infinite" number of acknowledgements, all related to the initial request. The number of acknowledgements will dependent on the type of communication service used:

a) Non-acknowledged write will not give an acknowledgement.
b) Ordinary function, read and write type transactions will give exactly one acknowledgement.
c) Subscribe type transactions give any number of acknowledgements. This also applies to broadcast subscriptions.

NOTE - Anonymous broadcast subscribe type objects are not transaction oriented, but are defined to be in the broadcast subscribe class for simplicity.

5.2.6 Reliable transfers

Non-broadcast transactions are reliable. Messages are guaranteed to reach their destination unless a connection error occurs. All connection errors will be detected within a prescribed time limit and reported to the client and server. A connection error means that some unacknowledged messages may have been lost, but the protocol cannot say how many or whether they were processed by the server.

Although broadcast transfers are unreliable, there is still reliability in the connection between sender and receiver. If the (unreliable) link to the server goes down, the client will be notified.

5.2.7 Real-time properties

The system provides three priority levels: Urgent, Normal and Low. Urgent messages will have priority above normal which have priority above low. Higher priority messages will get network bandwidth before lower priority messages.

Dependent on the T-profile in use, this means that it is possible to allocate a certain part of the total bandwidth to higher priority messages. This can be used to guarantee an upper bound on message latency for the different priority classes.

NOTE - The different T-profiles must be expected to support priorities in different manners. Some T-profiles may not support priorities at all. This will be defined by the relevant specifications.

5.3 Application management

5.3.1 MAU states

The MAU goes through certain states as a component in the system. The most important of these are:

a) Not connected. No connection with LNA - stand alone mode. The application can be operational, but it does not have connectivity to the network.
b) Connected: Connected to LNA and able to communicate with other MAUs.

The LNA provides certain services to manipulate and supervise these states. The LNA will also notify all servers and client MAUs associated with a dead MAU about the death.

5.3.2 System management

The T-profile will define certain mechanisms that allows a (possibly duplicated) system management function to be implemented. These mechanisms will depend on the T-profile in use. These mechanisms will cater for physical network management on the T-profile level, e.g., network load in terms of bits or octets per second, physical nodes and corresponding network address, T-profile level network errors etc. These services are defined in the T-profile document.

All LNAs will have a special MAU associated with it that allows the host computer or a central system management function to access statistics about the LNA. Certain management functions will also be available. The mechanisms implemented by this MAU will be related to logical (A-profile level) management functions. This includes statistics and general information about local and remote MAUs, data objects and transactions.

5.3.3 Time distribution

The T-profile will define services by which the host computers can synchronise their internal clocks to each other. The applications shall in turn use the host computer's time keeping function for synchronisation to the system.

The A-profile will not define time keeping functions beyond this service.

5.3.4 Load limitation

The LNA will provide a mechanism by which the communication load on the application unit can be limited. This mechanism will be based on specified upper limits in the interface connection requests. Such load limits can be specified both by server and client.

5.4 Data object connection management

5.4.1 Data object states

The states of each MAU's connection to a data object is reflected in the state of the MAU's MCP. The states are the same as for the MAU: Not connected and connected. The meaning of the states are a bit different for server and client MAU, but basically the connected state is a prerequisite for communicating.

Broken connections on the server side will be propagated to all client MCPs reflecting the relevant data object. This is handled by the LNA through the protocol.

5.4.2 Server object definition

A MAU can define new server objects at any time. The server does this by creating a local MCP for the served object and exporting the attributes to the LNA. The attributes are:

a) The server MAU name (implicit in the request).
b) The data object name (possibly divided into an interface and an object part).
c) The data structures used for requests and/or acknowledgements.
d) The communication service provided (see next clause).

The server may specify that it shall be informed about any connection attempt from a client. This instructs the LNA to send information to the server each time a client tries to connect to the object. The information from the LNA will contain client MAU identity together with an optional client supplied "password". The server MAU may accept or deny the connection attempt. An additional input parameter gives an identification of the client MAU's session with the server MAU (see coming clause). This session identity will be the same for all connection points the client MAU is connected to.

Several data objects may be exported as one interface. The functionality is the same as described above, but the data object names will have a prefix consisting of the interface name.

5.4.3 Client object connection request

A MAU can request a connection to a remote data object at any time. This is done by defining a client MCP and exporting the corresponding attributes to the LNA.

The object the MAU requests a connection to does not have to exist at the time of the request. The LNA will repeatedly try to connect to the object, until the connection attempt times out (time out may be set to infinite).

The LNA will inform the client MAU when the connection is established, if an error occurred, if the server denied the connection or if the connection attempt times out.

Several data objects may be connected as one interface. The functionality is the same as described above, but the data object names will have a prefix consisting of the interface name. A client side interface must be a sub-set of the server side interface, but they do not need to be identical.

5.4.4 Client MAU authentication

The LNA will use T-profile functionality to monitor its connection to other LNAs, which in turn monitor their connections to local MAUs. This monitoring will be used to verify that a connection between a local and remote MAU (carrying traffic related to one or more data objects) is not broken by possibly "hostile" MAUs or LNAs.

The local MAU can, if requested, get a session identifier together with all connection and transaction requests from remote MAUs. This session identifier can be used to check that different transactions and connection attempts come from one and the same remote MAU. This means that one (companion standard supplied) authentication mechanism can be used to cover many different data objects or transactions by letting the server verify that the session identifier do not change.

NOTE - The quality of this service will depend on the quality of the corresponding services in the T-profile.

5.5 Message transfer

5.5.1 Transaction states

A client MAU can initiate several concurrent transactions on one MCP. The transaction acknowledgment can also be delayed by the server. For this reason it is necessary to treat each transaction as an object. Each client initiated transaction will be reflected in a transaction object both on the client and the server side. The objects are created as a result of the client action and will be deleted as soon as the transaction is completed, times out or is cancelled.

5.5.2 Basic transaction principles

Message exchanges between client and server MAUs are based on a transaction principle. Each request from a client is assigned a unique transaction identity number which is returned to the client on completion. This means that:

a) The client can have any number of transactions pending.
b) The client can define a time-out for a transaction.
c) The server may service transactions in any order as long as the transactions identities are kept track of.
d) One subscription request is assigned one transaction identity. Repeated subscription messages will have the same transaction identity.
e) It is possible to cancel transactions and also subscriptions.
f) All transactions are automatically cancelled if the associated connection is broken.

Each transaction can result in the exchange of any number of messages, from zero (if an exception occurred on the client side) to infinite (for long duration subscriptions). All messages can carry a data structure defined by the applications (as an attribute in the data object identity). The format of this data structure is fixed and defined by the structure attribute.

The maximum size of the message is dependent on the T-profile. Unsupported length specifications, due to limits in the T-profile, will normally result in the denial the establishing of a connection to the data object in question.

5.5.3 Transfer mechanisms

The following transfer mechanisms are defined:

a) `READ, WRITE, FUNCTION`: Request the sending and/or receiving of one message. Reliable service. The write function is acknowledged.

b) `NON-ACKNOWLEDGED WRITE`: As normal write, but without acknowledgement. The service is reliable, but one cannot know how many messages were lost if a connection failure occurs.

c) `SUBSCRIBE`: A number of clients subscribe to one server and get any number of messages back from the server. The server decides when a message is sent. All clients receive the same message except for the first message returned from the subscription request. This message is only sent to the subscribing client. The service is reliable.

d) `BROADCAST SUBSCRIBE`: As normal subscribe, except that messages are sent as broadcast datagrams. This means that messages (except for the first message after subscribe request) can be lost. The connection management, however, is based on a reliable LNA-LNA link. A set of pseudo-objects will be defined that allow MAUs to broadcast and/or listen to globally defined connection points.

e) `INDIVIDUAL SUBSCRIBE`: A mix between function call and subscribe where the different clients may request individual subscription messages. The server needs to keep track of the clients itself.

f) `ANONYMOUS BROADCAST SUBSCRIBE`: This function is similar to ordinary broadcast except that there are no link between the sender and receiver of messages. Every MAU can send or listen to these messages.

5.5.4 Data marshalling

The protocol defines a hardware and network independent transmission format that ensures that individual data entities and structures are received with the same meaning as was sent. This mainly means that receivers and senders must convert between different byte order and structure element ordering and padding. The API will be required to handle this data marshalling. The following types of data elements are supported:

a) Signed integers: 8, 16, 32 and 64 bits of length.

b) Unsigned integers: 8, 16, 32 and 64 bits of length.

c) Characters: 8 and 16 bit lengths.

d) Floating point: 32 and 64 bits lengths.

In addition to individual data entities, the standard supports the following structuring:

a) Fixed length arrays of entities or records.

b) Variable length arrays of entities or records.

c) Records of entities and/or records and/or arrays.

d) Unions of entities or records. This is a type safe implementation where legal records or entities are enumerated and the relevant code is transmitted with the message.

e) Opaque octet blocks. These are unstructured blocks of octets that are given a meaning only through the use of special companion standard document called “information specifications”.

The maximum length of a data structure is given by the T-profile’s MTU limitation.

5.5.5 Authentication

All transactions will be tagged on the server side with the client MAU’s session identifier. This is done by the LNA based on transport level connection management services. The session identifier can be used to identify a MAU and a session across different server connection points. The standard will provide a companion standard for user and client authentication that can be used together with the session identifier.

NOTE - The quality of the authentication will dependent on the quality of the T-profile used to do the authentication.

5.6 Bulk transfer

5.6.1 Mechanism

The T-profile will provide a mechanism for transfer of large amounts of data over a low or normal priority communication link. The LNA will provide a service to the MAU where this mechanism is made available. The transfer will be made in one of two ways:

a) The MAU transfers the data block to the LNA with an IPC mechanism. The LNA transfers the data block to the destination.
b) The MAU passes a *reference* to the data block to the LNA. The LNA reads the data directly and transfers it to destination.

The receiving side will get data in a corresponding manner. The LNA will take care of flow control to avoid overloading the network or the involved MAUs.

5.6.2 Application level activation

This will be handled by a transferring the bulk addresses between sender and receiver MAUs through normal MCP activations. The companion standard will contain provisions for this.

6 T-profile functionality

6.1 Introduction

The T-profile shall supply a set of standard transport and network related services to the A-profile. This set of services may have different quality of service (QoS) for different T-profiles, but all shall be present as prescribed by the standard. The A-profile will be defined in terms of how these basic services are used, independently of what T-profile is in actual use.

The part of this standard specifying the general T-profile requirements, services and the quality of service attributes will also define two concrete T-profiles over redundant Ethernet and non-redundant Ethernet respectively. That part will also provide definitions of how to interconnect MAUs over the wide area Internet. The last definitions can be used for remote test integration or various forms of remote diagnostics and maintenance. Other future Parts of the 61162-4 series may specify other T-profile implementations.

6.2 General overview of quality of service

Different T-profiles will have different capabilities. Each T-profile specification shall provide a mechanism by which the capabilities can be inspected. Typical parameters that can be requested are:

a) Maximum transmission unit (MTU) size. The MTU will define an upper limit for message sizes for certain transport services, normally urgent and normal priority messages. The MTU shall be specified for both urgent and normal messages. Some T-profiles may not have a maximum MTU.
b) Priority levels actually implemented. The standard specifies three priority levels, but a given T-profile may support a higher or lower number of levels. Different numbers of levels may be available for the different message or stream services. The A-profile will be able to use priority levels low, normal and urgent.
c) Real-time properties. The T-profile should be able to report its support for timely delivery of data, possibly dependent on the priority class and the network and host computer load.
d) Support for stream service. Some T-profiles may not support the stream service.
e) Support for redundant transport network. Some T-profiles shall be able to supply redundant communication paths to the A-profile. This service shall, if available, be independent of A-profile.
f) Communication link authentication quality. The T-profile shall have the possibility to identify end-points of a communication link so that these cannot be changed during operation. The

quality of this mechanism will impact the corresponding authentication service in the A-profile.

g) Time precision related to a global reference. Some T-profiles may not implement a global time service.

h) Network management services. Not all network management services may be available for all T-profiles.

Other parameters related to the quality of service (e.g., maximum transmission delay, buffer sizes and various control parameters) may also be made available.

6.3 The T-profile services

The standard T-profile services include time distribution and network management services as well as data transport services, although the former are not part of the transport profile in the OSI sense. These services and protocols are included in the T-profile fragments due to their dependencies on the transport level protocols.

6.3.1 Network address look-up and mapping services

The T-profile shall be able to assign network specific addresses to the nodes. It shall also have provisions for mapping MAC addresses to network addresses. This mechanism shall make the A-profile independent on the MAC or network level addressing mechanism.

6.3.2 Reliable message service

Transport of finite length messages from point to point. This service shall be provided with the priority levels urgent and normal. Message length for these messages may be limited to maximum transmission unit (MTU) for the actual T-profile in use.

The service will be based on a connection oriented principle where connections are established prior to message exchanges.

6.3.3 Reliable stream service

Transport of stream data (octets) from point to point. This service shall be provided with the priority levels normal and low. Some T-profiles may not support this service.

This service will also be based on a connection oriented principle.

6.3.4 Unreliable datagram service

Multi- or broadcast transmission of datagrams. Message length may be limited. This service shall not depend upon connection being established as seen from the A-profile. As a minimum, the T-profile shall supply one multi-cast address that will be listened to by all LNAs in a network.

6.3.5 System management

The T-profile shall provide certain system management functions, e.g., with the help of SMTP. The services include network load monitoring and network node health monitoring. They may also include services for the allocation of network level addresses to nodes.

6.3.6 Time distribution

The T-profile shall specify how time is distributed and synchronised. The quality of service attributes should include a measurement of expected offset from global time, e.g., UTC.

6.3.7 Exception handling and reporting

The T-profile shall specify what failure modes it can detect (and possibly correct) and how these are reported. Exception reporting should normally go via the network management services.

7 Companion standards

7.1 Introduction

The main purpose of the IEC 61162-4 Series companion standards is to provide an unambiguous way of interpreting data transmitted via the protocol. In this sense, the companion standard adds meaning to the data, converts it into information and makes it useful for all applications connected to the network. To serve this purpose, the companion standard defined in Part 420 of this standard provides the following:

a) A language used to define information types, data types, application interfaces and applications. This language is readable as well as interpretable by a computer.

b) A standard set of information types, application interfaces and applications making up part of the PISCES Foundation Specifications (PFS). This set is used to create applications within the framework of this standard.

c) A way to structure applications within the PFS framework. This is done by making certain parts of the PFS components mandatory and by defining a hierarchical tree for construction of applications from non-mandatory components.

d) A framework for creating new applications from the PFS and document them with the Companion Standard language.

The standardised set can be extended by the application programmer, using the companion standard language and rules. The set of components (particularly the PFS) can also be extended in the future by adding new documents to this standard.

7.2 The companion standard functionality

The companion standard documents (the PFS and the final application documentation, except for the information type documents) represents the specification of how certain protocol level entities shall be created to establish certain application dependent communication links. The entities are basically the ones used as attributes to the data objects:

a) The MAU name (for both server and client)

b) The interface name.

c) The data object name.

d) The application data structure for the transaction messages.

e) The communication functionality supplied by the data object.

These attribute values will be unambiguously defined through a complete set of companion standard specifications.

In addition, the companion standard will allow application programmers to define “information specifications” that give further instructions in how to interpret the data transported between applications.

7.3 The companion standard language

The application standard language description defines four types of documents:

a) Data type definitions. These documents define the structure of data structures used in messages.

b) Interface definitions: These documents define the construction of data objects and how the data objects are organised in interfaces.

c) Application specifications: These documents define the construction of actual applications and resolves configurable name issues and assigns server and client functionality for interfaces to the application.

d) Information specifications: These documents provide specifications on how to interpret data transmitted in messages between clients and servers. These documents can also in some cases provide additional structuring information for data (if the data is of the opaque block type).

The companion standard general principles fragment provides this information.

NOTE - It is possible to create applications using this protocol that cannot be described by the companion standard language. The language must be viewed as representing a subset of the possible functionality and that this subset has been selected to give a better structure to applications using this protocol.

7.4 Companion standard PFS components

The companion standard general principle will also define certain components that can or must be used by PFS compliant applications. These components are:

a) General version control and manufacturer related information retrieval.

b) Interface for general data access in alarm and monitoring systems of any type.

c) Interface for transmission of IEC 61162-1 telegrams over this protocol.

Other Parts will define other standard application areas.

7.5 Companion standard PFS structure

The companion standard will also define a certain way to include standard interface components for applications of various classes. This serves to create a general framework for applications that allow general system management functions to be implemented independently of the supported functionality.

7.6 Companion standard application description

Finally, the companion standard general principles will provide a standard way to document application interfaces so that these can be understood in an unambiguous way by any manufacturer, integrator or owner knowing this specific syntax.

8 System configuration services

8.1 General

MAUs and LNAs are designed for a wide range of applications, thus being characterised by a number of configurable parameters. The configuration services offered by this standard define these parameters and the way to modify them at module and system levels.

8.2 System configuration principles

An IEC 61162-4 system shall consist of a number of nodes exchanging information according to the client/server scheme. It is important to build and maintain such a system with minimal effort, a fact that requires a lot of automatic configuration as well as operator-controlled configuration. API services are provided to enable such changes in configuration.

Generally, configuration information may be saved in a non-volatile memory such that when a node is removed and reinserted (either the same or a compatible one), all parameters are restored (time synchronisation, active connections, open files). This issue is addressed at the companion standard level.

Parameter changing may be done without stopping the application. Application software upgrades may be done by passing functionality to a backup MAU (if necessary), stopping the application and restarting it with the new software. Both may be performed either locally (through serial port or an attached console) or remotely (over the ship network).

8.3 Physical network configuration

Normally the system integrator needs to configure the physical network parameters. This is done by assigning physical network addresses to the relevant nodes after a template has been defined by the system integrator. It is possible to do this automatically by utilising Internet protocol services, but this is not recommended by this standard.

The standard prescribes the use of suitable network management protocol to supervise error conditions and traffic status on the physical network level. various T-profiles will prescribe different protocols. The system integrator must normally provide a network monitoring station as part of the network configuration.

8.4 Application configuration

Most MAU parameters can and should in principle be defined from the factory. The following list identifies the most important parameters and indicates those that may have to be set by a system integrator:

a) MAU name. This must be unique in an installed system and may not always be possible to define without knowing the actual installation.
b) Server MAU names as used by the client MAUs, in the cases where the configuration noted in 1. above possibility is used.
c) Password where this is used to restrict access to the MAU (e.g. to allow authorised users only to access the MAU or change the configuration).

8.5 Error monitoring and reporting

Errors may be application, communication, or configuration errors. It is essential to report when a source of information is lost even if no data is expected at the time of failure. The API provides services which monitor such errors, log the events, provide for MAU notification and enable remedial actions (disconnection, re-initialisation, configuration change) according to their severity. The programmer may design the actual approach to tackle the possible problems.

In case of failure of a remote server application, a backup MAU must be sought. If this is not possible, the client applications must monitor when the server is up and running again. If the failed remote side is a client, a warning is issued. It is possible though that the failure is in the communication link. Then, a redundant path must be followed. If this is not possible, the LNA shall attempt to connect to the other side as soon as the link is operational again. In all cases the active LNAs update their internal databases of live connections and continue to listen for other LNAs' activity. Such diagnostic functions on the network must not delay safety-critical data.

The standard prescribes the use of a separate network management protocol to implement these functions. This protocol will be different for different T-profiles.

8.6 Load/Performance monitoring and reporting

It is primarily the role of a network tool to monitor network activities such as link utilisation, packet delays and errors.

However, the LNAs that constitute the network segment of an IEC61162-4 system, may well monitor network traffic that refers to transactions, connections and related events. Such information may be logged and viewed off-line.

Additionally, LNAs may interface to the network protocols in order to acquire additional information such as link utilisation and timeliness. These will help to apply measures such as a flow control and rate control to the MAUs attached to the LNAs, or initiate a degraded mode of operation if the network performance is deteriorating. This functionality may be performed on-line.

The standard prescribes the use of a separate network management protocol to implement these functions. This protocol will be different for different T-profiles.

8.7 System inspection and configuration management

It is possible to inspect on-line a MAU or an LNA in order to obtain restricted information such as version codes, MAUs served, MCPs and interfaces in use and active LNAs in the network. This can be done by using functionality defined in the A-profile and in the companion standards.

Annex A
(Normative)

Typographical conventions and nomenclature

A.1 Use of typeface

This standard will use the `courier` typeface for symbolic constants and data types that are defined to have special meaning here. All other text will use the normal typeface.

A.2Regular pattern

This standard will use regular patterns to formally define the format of certain text strings. This means that the string format will be presented as a skeleton string where certain characters can be substituted with zero or more other characters. The following general rules will be used in this standard:

a) All characters except '[', ']', '+', '*', '?' and '\' represent themselves.

EXAMPLE - "abc" defines a string with exactly that form, i.e., "abc".

b) Square brackets ('[' and ']') enclose a list of characters. This construct represents one character from the specified list. The dash ('-') inside square brackets is used to identify a range, i.e., "a-f" shall be interpreted as "abcdef".

EXAMPLE - "[a-c]d" defines three different formats for the string in question, i.e., "ad", "bd" and "cd".

c) The star ('*') means that the preceding character can be repeated <u>zero</u> or more times.

EXAMPLE - "[ab]*" defines infinitely many strings, e.g., "" (the empty string), "a", "bba" and so on.

d) The plus sign ('+') means that the preceding character can be repeated <u>one</u> or more times.

EXAMPLE - "[ab]+" cannot represent the empty string "".

e) A question mark ('?') represents any legal character [ISO 8859-1], including non-printing characters.

EXAMPLE - "?*" represents any string, including the empty string.

f) A back-slash ('\') means that the immediately following character shall be taken literally, except when the three following characters are octal digits specifying a legal character code as defined in [ISO 8859-1].

EXAMPLE - "\?" defines the string "?", "\015" represents a "carriage return" control character.

In some cases these definitions will be extended with other characters with special meaning.

A.3 Constant representation

Table 2 defines the allowed ways to write literal constants in this standard.

Table 2 - Constant representation

	Form	Example	Description
1	[0-9]+	1, 23, 456	Positive integer in decimal notation
2	0x[0-9a-f]+	0x1, 0x12c, 0X12C	Positive integer in hexadecimal notation
3	0[0-7]*	01, 072, 0454	Positive integer in octal notation
4	-[0-9]+	-300	Negative integer in decimal notation
5	N.F, NeN, N.FeN	1200.0 12e2, 1.2E3	Floating point
6	'?'	'å','b','\023'	One character
7	"?*"	"text", "\013\012"	Text string

	Form	Example	Description
8	{E, E, ...}	{“text“, ‘b’, {32, 3}}	Aggregate (array or record)

The *form* column defines the legal format of the literals as an extended regular pattern. The extension is the introduction of the special characters ‘N’, ‘F’ and ‘E’:

a) An ‘N’ represents a positive or negative decimal number as defined in line 1 or 4.

b) An ‘F’ represents a positive decimal integer as defined in line 1.

c) An ‘E’ represents any other literal as defined in items one to ten, i.e., recursion is allowed.

Each of the following items explain the corresponding line in the table:

a) This standard uses a decimal representation for all values unless stated as in clauses below. Decimal numbers consist of any sequence of the digits zero (0) to nine (9).

b) Hexadecimal notation is used in some cases and is always identifiable by the prefix 0x immediately before the hexadecimal number, e.g., 0x100 is equivalent to 256. Legal digits in hexadecimal numbers are 0-9, a-f and A-F. It is also allowed to use the prefix 0X.

c) Octal form is recognised by the prefix zero (0). Legal digits in octal numbers are 0 to 7.

NOTE - A literal with a leading zero, but which includes the numbers 8 or 9, is interpreted as a decimal number. A leading minus means that the number must be interpreted as a negative <u>decimal</u> number.

d) Negative integers are indicated with a minus sign in front of a <u>decimal</u> number.

e) Floating point numbers shall be written as in line five of the table. An ‘e’ (or optionally an ‘E’) prefix the exponent. Negative mantissa or exponent is indicated by a minus sign directly in front of the respective number. Only integer exponents of ten are allowed.

f) Characters are put inside single quotes as shown in line six. Character and string definitions may contain “\NNN“ to represent non-printing (and printing) characters. Exactly three octal digits give the numeric code of the character as defined in [ISO-8859-1].

NOTE - Literal text strings of the form “\ooo“, where ‘o’ denotes a legal octal digit must be written as “\134ooo“. “\134“ is this standard’s special code for back-slash.

g) Text strings shall be written within double quotes as shown in line eight.

h) Literal records or arrays are written as in line ten. The number and types of elements must match those of the record or array it is associated with. It is legal to use nested array or record literals.

A.4 State machine descriptions

Services that the protocol supplies to the applications are in places described in the form of state transition diagram. Figure 8 shows parts of a state transition diagram.

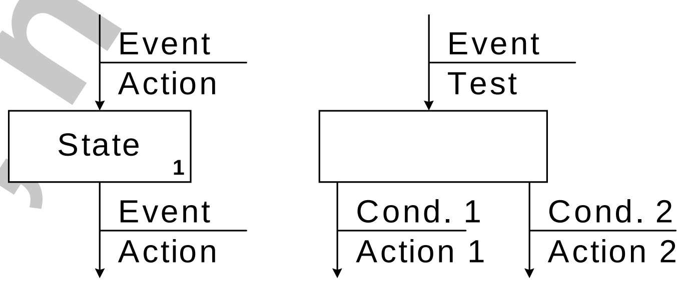


Figure 8 - State transition diagram nomenclature

The left-most figure shows an ordinary state (rectangle) with transition in and out. The name of the state is inside the rectangle and there may also be a state number in the lower right corner. Arrows pointing into the state show transitions to the state and arrows leading out of the state show transitions to other states.

Each transition is labelled with the event (above the bar) that caused this transition and the action (below the bar) that will be taken on entry to the new state.

The right-most figure shows a test. The in-going transition arrow has a test instead of an action and the outgoing arrows has a condition instead of event. No state name or number are associated with tests.

A.5 Context diagrams

Context diagrams are used to describe the environment a given module operates in. An example is shown in Figure 9.

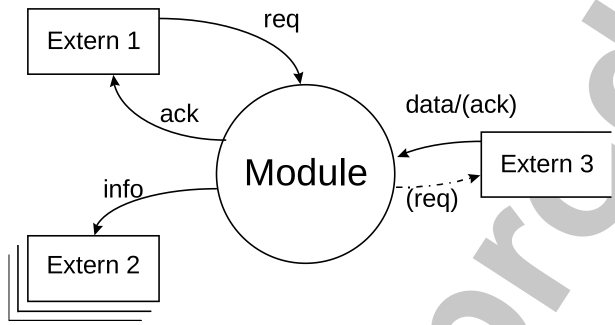


Figure 9 - Context diagram nomenclature

The described module is shown as a circle, communicating entities as rectangles. A set of identical instances of communicating entities are shown as a stack (*Extern 2*). The arrows show communication flow between entities, dashed arrows show communication exchanges that are not required by the standard.

The communication between the module and external entities are done by exchanging messages. The messages are generally of the following types:

a) Request/acknowledge exchanges (*req/ack* in the context diagrams) where one module requests a service and the other supplies (or denies) it.

b) Data driven exchanges (*data* in the context diagrams) where messages from one module shall be processed and possibly sent on to another. The sender relies on the message being processed by the receiver or that an exception is raised (usually indicated by another data message in the reverse direction).

c) Informational messages (*info* in the context diagrams) where one module sends an unacknowledged information message to another and do not rely on it being processed or re-transmitted.

A.6 Entity-relationship (ER) diagrams

This standard uses ER (entity-relationship) diagrams to specify relationships between objects. The standard uses the OMT (Object Modelling Technique) nomenclature where an object is an instance of a class. In an entity relationship diagram, the object model in OMT, each class represents zero or more objects in the system. The objects represents physical units, information entities or any other physical or abstract item that have relationships to other items. The different relationships are described in Figure 10.

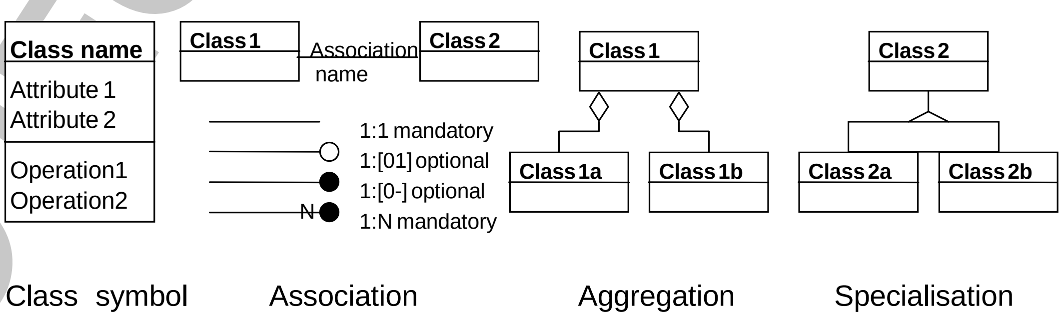


Figure 10 - ER-diagram nomenclature

From left to right is shown examples of:

a) **Class symbol**: Classes that represent objects are drawn as shown or simplified as on the right hand side. In bold type-face at the top is the class name. Then (optionally) a list of attributes (e.g., colour, size) and then (optionally) a list of operations (e.g., connect to, send, receive).

b) **Relations**, the multiplicity e.g. one to many can be specified, and Associations: Named relationships between two objects.

c) **Aggregation**: A Class 1 object consists of (or can be decomposed into) one Class 1a object and one Class 1b object.

d) **Specialisation**: Class 2 can be specialised into Class 2a and Class 2b. A Class 2a object has all the attributes and operations from Class 2a, but it also inherits all the attributes and operations from Class 2.

Restrictions

In this standard, object orientation is only used to model systems. To avoid confusion with C++, no classes, objects or instantiations shall be mentioned.

A.7 Structure of service descriptions

This standard describes a set of protocols (T-profiles, A-profile and possibly some companion standard protocols). These protocols require the implementation of a set of services to make the functionality of the protocol available to programmers. This section describes the standard format for the service descriptions used by this standard.

Simple object orientation with events

The service descriptions will use a very simplified object orientation combined with an event driven programming paradigm. This principle is chosen because it maps well to the state transition diagrams used in the description of the protocols and because they can be implemented independently of most types of hardware and operating system restrictions on possible host computer systems.

The general object and event based concept has been chosen to create a consistent service description, but they are not mandatory for a new implementation of the protocol or parts of the protocol. The modularization of the protocol itself allows different programming paradigms to be used in different parts of one host computer's program and the message based nature of the system will also allow different paradigms to be used in different host computers.

The service descriptions will be focused on certain objects, for instance a communication port, a communication buffer, an application, a protocol connection point or a transaction.

The service descriptions will then be based on the application programmer's need to change the state of the object, e.g., to initiate a connection attempt and the protocol implementation's need to notify the application program of changes in object state that is caused by external events. Both application program service invocations and notification of state changes can be looked at as events to the object.

Services that can be invoked by application program

These services will normally be implemented as a set of sub-routines or "methods" that can be invoked by the application program. In the descriptions they will be given a name consisting of upper and lower case letters. This name will be included in the clause heading, in parenthesis. The service names may be used in state transition diagrams to indicate user and then always type set in `courier`.

Call back to application program

Call back services are included as separate clauses, but with the name of the application service followed by the words “call back” in the clause heading.

Call-back descriptions use the same format as normal service descriptions, but with some of the fields marked with not applicable.

Call back parameter list

Call backs can be created by the program designer in many ways. In general it is recommended that the call back has at least three parameters:

a) The identity of the object the call back refers to.
b) The identity of the event that caused the call back.
c) An application program defined identifier (defined when the call back is first registered, usually when creating the object in question) that can be used for application level call back identification.

This structure is flexible enough to allow also object oriented principles to be used in a procedural language. For object oriented implementation it may be natural to look at the call-back as a method and, thus, it would be unnecessary to include the first parameter since that is defined by the method the object was invoked on.

Service description format

All services will be described in a fixed format. This format consists of the following fields:

a) Name and short explanation: In the clause title
b) Explanation: First paragraph in clause body.
c) Pre-condition: Required state or actions before service in invoked.
d) Post-condition: result of service completion in terms of system status.
e) Input parameters: List of necessary input parameters.
f) Output parameters: Same for outputs from service invocation.
g) Call-back: Specification of the condition for any call-back (up-call).
h) Errors: Specification of possible errors that can occur.

Example of service description (ExampleService)

This is an example description. This paragraph would contain a short description of the service which in this case is called *ExampleService*.

Pre-condition: Necessary prerequisites for invoking service.

Post-condition: Change in system state after service has been completed.

Input parameters: As described.

Output parameters: As described.

Call-back: Any call backs that can result.

Errors: Any error returns from the service itself (other exceptions can be listed under call-back).

Example call back description (ExampleService call back)

Pre-condition: As for previous.

Post-condition: As for previous.

Input parameters: As for previous, note however that object identity and event codes are normally left out of this list..

Output parameters: Usually not applicable.

Call-back: Not applicable.

Exceptions: Usually not applicable.

____